第31卷 第23期 计 算 机 工 程 2005年12月
Vol.31 № 23 Computer Engineering December 2005
·基金项目论文· 文章编号：1000—3428(2005)23—0033—03 文献标识码：A 中图分类号：TP393.08

数字签名技术在办公自动化系统中的实现

阳文泽，李翠华
(厦门大学计算机科学系，厦门 361005)

摘 要：开发了一种基于密钥集中式管理的数字签名方法，并在开发的办公自动化系统中予以实现。该方法利用 MD5 算法来抽取公文信息指纹，利用 RSA 公开密钥算法在公文信息指纹上实施签名，从而保证了公文信息的完整性，并通过一种密钥生产管理系统来生产并管理办公自动化系统中所有合法用户的公/私密钥对，从而保证签名的真实性与安全性，并为用户使用密钥提供方便。通过测试表明，该文实现的数字签名具有良好的运行性能，完全可以满足在办公自动化系统下运行的要求。

关键词：数字签名；密钥管理；消息摘要

Implementation of Digital Signature in Automatic Office System

YANG Wenze, LI Cuihua
(Department of Computer Science, Xiamen University, Xiamen 361005)

【Abstract】 A novel digital signature method is implemented in the automatic office system. In this method, MD5 information digest algorithm has been adopted to extract the print of documents and RSA public key algorithm has been adopted to produce digital signature on document print. These two algorithms can protect documents from being tampered and document receiver can make sure of the author of the document he received. In order to make sure the key's origin and security, the paper comes up with a subsystem called the key build and management system to produce keys for all valid users. The running test proves that the method has an ideal performance in measuring of executing time and memory space.

【Key words】 Digital signature; Key management; Information digest

1 概述

公文的创建者需要对公文签名以表明其创建者的身份。根据签名，可以确定某份公文是某个办公主体所创建的，并对公文内容承担责任。这个过程需要考虑两个问题：(1)文档创建者如何保证签名文档的完整性？即签名 S 所在的文档是没有被篡改的；(2)文档接收者如何辨别签名的真实性？即签名 S 是属于办公主体 A 的而不是另一个办公主体 B 所伪造的。在电子信息的环境下，文档数据的特征在相同编码方式的前提下是由其实际内容来决定的，文档内容的编码可以进行压缩而不丢失文档特征，这种压缩过的文档内容编码称为这个文档的数据摘要，如果抽取数据摘要的过程不可逆，即由数据摘要不能推导出原数据文档，那么文档的创建者可以将文档与文档摘要同时发送，而文档的接收者对接收到的文档使用相同的摘要抽取算法，并比较接收的文档摘要和自己所抽取出的文档摘要，如果它们相同，则表示文档是没有被篡改过的。目前已经有比较成熟的数据摘要抽取算法，如 RSA 公司的 MD2、MD4、MD5 算法[1]以及 NIST 在 1993 年 5 月公布的 SHA 算法[3]。一般而言，这些算法的安全顺序为：MD2<MD4<MD5<SHA，运算效率则相反，选用 MD5 算法可以同时兼顾安全性和高效性。本文选用了 MD5 算法作为在办公自动化系统中抽取数据摘要的方法，并讨论其具体的实现过程。

目前基于公开密钥算法的签名方法已被推荐为数字签名的标准[2]。公开密钥算法是一种不对称的加密方法，即加密密钥和解密密钥不相同，但存在一一对应关系。加密方持有的密钥称为私钥，解密方持有的密钥称为公钥。文档创建者 S 使用私钥对要传输的文档进行加密，文档接收者 R 使用公钥对接收的文档进行解密，由于由公钥来推算私钥是一个 NP 问题，因此 R 也就无法冒充 S 在篡改接收到的文档后，再将其加密发送给第 3 方。在数字签名的过程中，一般由 S 来创建公/私密钥对，并由 S 向 R 公布其公钥，由于公/私密钥之间存在一一对应关系，因此如果 R 确认两点：(1)使用 S 的公钥可以对某个文档 D 进行解密；(2)公钥确实是属于 S 的，那么 R 就完全有理由相信文档 D 的确是由 S 发送的，并且 S 也无法否认这个事实。对于 R 需要确认的第(1)点，只需执行解密过程便可得到结果；对于第(2)点，或者是 S 在创建完自己的公/私密钥对后，亲自将公钥交到 R 手中，或者是有一个第 3 方仲裁机构，它负责登记 S 的信息并保管、发布其公钥，R 则应是充分信任这第 3 方的权威机构，并从它那获得 S 的公钥，公钥的真实隶属关系完全由这第 3 方来保证。在办公自动化系统中，办公的主体很可能不是处于同一个地域，而且在自动化的环境下还需手工的交换密钥是一很不合时宜的事情，因此由 S 亲自将密钥送到 R 手中的方法不可行；提供第 3 方的担保需要切实保证第 3 方的权威性，如数字证书[5]方法，在办公自动化环境中，我们提出了密钥生产管理子系统的概念，以此作为公钥交换的第 3 方。密钥生产管理子系统负责为每一个合法的办公主体创建公/私密钥对，负责向 S 提供其私钥，向 R 提供 S 的公钥。密钥生产管理子系统提供了密钥集中式管理的方法，使得密钥的创建与用户无关，避免

基金项目：国家创新研究群体基金资助项目(60024301)；国家自然科学基金资助项目(60175008)；福建省自然科学基金资助项目
作者简介：阳文泽(1974—)，男，硕士，研究方向：计算机信息安全；李翠华，博士、教授
收稿日期：2004-10-13 **E-mail：**lunwen_oywz@126.com

了用户制造虚假密钥的风险，也使得密钥的管理更加方便、更加安全。本文将就这一子系统在办公自动化系统中的实现作详细的论述。

2 MD5 算法及其实现

MD5 算法[4]将任意长度的信息映射成为 128 位长的值，作为这段信息的“摘要”。算法使用了 4 个基本逻辑函数：

(1) $f(a,b,c)=(a\&b)|(\sim a\&c)$; (2) $g(a,b,c)=(a\&c)|(b\&\sim c)$;

(3) $h(a,b,c)=a\wedge b\wedge c$; (4) $i(a,b,c)=b\wedge(a|\sim c)$。

每个逻辑函数的输入都是 3 个 32 位的变量，进行函数定义的位运算之后，输出 1 个 32 位的结果。

算法的处理过程如下：

(1)增加填充位

令原始信息长度为 L，填充信息的长度到 L'，使得 $L'\bmod 512=448\bmod 512$；填充方法是在实际的信息之后填加 1 个 1 位和若干个的 0。

(2)填充长度

用 64 位表示实际信息的长度，并将其附加在(1)所得结果之后。经过这步的处理信息的长度变为 512 的整数倍。

(3)将原信息按 512 位来分组处理

对每一个 512 位的分组，将其进一步划分成以 32 位为单位的子分组，这样可得到 16 个子分组，不妨设为 m[16]；

初始化 4 个 32 位的变量 a，b，c，d：a = ox01234567；b = ox89abcdef；c = oxfedcba98；d = ox76543210 并将其初始值保存到另外 4 个变量中 (a',b',c',d')。

对于每一个 512 位的分组，要进行 4 轮(对应 4 个逻辑函数)的处理。

在每一轮中有 4!次逻辑函数调用，因为任取 a, b, c, d 中的 3 个作为函数的输入共有 4！种取法。然后将逻辑函数调用所得结果加上第 4 个变量、信息的一个子分组和一个常数，再将所得结果向右环移一个不定的数，并加上 a、b、c 或 d 中之一，最后将结果值赋值给变量 a、b、c、d 中的一个。

进行完 4 轮处理后令 $(a',b',c',d')=(a',b',c',d')+(a,b,c,d)$ 对所有 512 位分组都进行如上过程的处理，处理完成后将输出的结果 (a',b',c',d') 作一个连接就是信息的摘要，每个变量都是 32 位，共 32*4 = 128 位。

算法实现中的问题：

问题 1 对于一个给定的文档，首先计算其长度 L，由库函数计算出的 L 表示的是文档中字节的个数，而算法中需要的是以位为单位的长度，即 L×8，这个数值很可能超出一个整型类型(即 4B)所能表示的范围，而且算法中要求使用 8B(即分组填充后的最后 64 位)来表示这个长度。

解决方案：L 乘以 8 的操作可以通过将 L 左移 3 位的操作来实现，但仅仅通过左移可能会丢失进位，即发生溢出；这里使用两个整型变量 Hnum、Lnum 来记录 L×8 的结果，计算过程如下：

Hnum = L >> (32 - 3)；Lnum = L << 3

Hnum 保留了 L 左移时可能溢出的位，如果没有溢出 Hnum 就是全 0；然后将 Hnum、Lnum 分别拷贝到最后一个分组的最后 8 个字节的位置上即可满足算法要求。

问题 2 由于计算机是以字节为单位来读取文档中的数据，而算法每次处理的分组长度是 512 位，即 512/8 = 64B，因此设置一个 64B 大小的缓冲区 inBuf 用于存放每次处理的分组数据。算法中给出的位填充方式是使得经过填充后信息的长度 L' mod 512= 448 mod 512；由于每次取 512 位数据，因此只在最后一个分组中可能出现需要填充的情况，根据最后一个分组的长度，我们需要考虑几种相应的填充策略。

解决方案：由于最后一个分组的后 64 位要用于存放消息的长度，因此以 512 – 64 = 448 作为分界点，那么最后一个分组的长度 length 就可能有以下 3 种情况：

(1)length < 448：这种情况下可使用算法中给出的位填充方式填充分组的长度到 448，然后再加上信息长度位；

(2)length = 448：这种情况不需要填充位，直接加上信息长度位即可；

(3)length > 448：这种情况下需要新创建一个分组，设当前的分组为 A，新建的分组为 B，那么对 A 中不足的位按算法给出的位填充方式进行填充，对分组 B，将其前 448 位全部填充 0，然后加上信息长度位。

3 RSA 公开密钥算法原理

RSA 算法是基于欧拉定理的一个重要推论：给定两个素数 p 和 q，整数 n = p× q 和 m，其中 0 < m < n，则下列关系式成立：$m^{\phi(n)+1}=m^{(p-1)(q-1)+1}\equiv m\bmod n$。其中 $\phi(n)$ 是欧拉函数，即小于 n 且与 n 互质的正整数的个数，对质数 p 和 q，有 $\phi(pq)=(p-1)(q-1)$。

算法对于明文分组 M 和密文分组 C，加密和解密的过程如下：

$$C=M^e\bmod n$$

$$M=C^d\bmod n=(M^e)^d\bmod n=M^{ed}\bmod n$$

其中收发双方均已知 n，发送方已知 e，只有接收方已知 d，因此公钥为{e，n}，私钥为{d，n}。该算法在使用过程中需要满足以下条件：

(1)可以找到 e，d 和 n，使得对所有 M < n，有 $M^{ed}\equiv M\bmod n$；

(2)对所有 M < n，计算 M^e 和 C^d 是比较容易的；

(3)由 e 和 n 确定 d 是不可行的。

对于条件(1)，由上述欧拉定理推论，如果令 $ed=\phi(n)+1$，则可得 $m^{ed}\equiv m\bmod n$，即可满足条件，而 $ed=\phi(n)+1$ 等价于 $ed\equiv 1\bmod\phi(n)\Leftrightarrow d\equiv e^{-1}\bmod\phi(n)$，即 d 和 e 是模 $\phi(n)$ 的乘法逆元。根据模算术的性质，仅当 d 和 $\phi(n)$ 互素(因此 e 也与 $\phi(n)$ 互素)时，d 和 e 是模 $\phi(n)$ 的乘法逆元。

对于条件(2)，由于 M 和 C 一般都是大整数，如果先对其进行幂运算，然后再对 n 取模，那么会导致中间结果非常大，这里可以利用模算术的性质进行模幂运算的转换：

$$[(a\bmod n)*(b\bmod n)]\bmod n=(a*b)\bmod n$$

假定要计算 Me，首先将 e 表示为二进制数，即 $e=\sum 2^i$，则可得到下面的结论：

$$M^e=M^{(\Sigma 2^i)}=\prod M^{(2^i)},M^e\bmod n$$

$$=\prod M^{(2^i)}\bmod n=(\prod[M^{(2^i)}\bmod n])\bmod n$$

对于条件(3)，由 e 和 n 确定 d 的难度取决于或等价于由 n 分解出素数因子 p 和 q 的难度，如果素数因子 p 和 q 取得太小，那在有限的计算时间之内由 e 和 n 来确定 d 是有可能的，因此条件(3)实际上等价于大素数的生成问题。目前还没有有效的方法可以产生任意大素数，通常采用的方法是随机选择一个期望大小的奇数，然后测试它是否是素数，若不是，则选择下一个随机数直至检测到素数为止。

下面给出 RSA 算法的一般实现过程：

(1)随机地选择两个足够大的素数 p 和 q；

(2)计算上述素数的乘积 $n=p*q$；

(3)计算 n 的欧拉函数值 $\phi(n)=(p-1)(q-1)$；

(4)随机选择一个与 $\phi(n)$ 互素的整数 d，令 SK = d；

(5)解同余方程，$SK*PK\equiv 1\bmod\phi(n)$ 求出 PK。

由上述 5 个步骤可得到 RSA 算法所需的公钥{PK，n}以及私钥{SK，n}。

(6)加密过程：$Y = X^{SK} \bmod n$，其中 X 是明文，Y 是密文，并且 $X, Y \in (0, r-1)$；

(7)解密过程：$X = Y^{PK} \bmod n$，X 与 Y 的含义同加密过程。

4 密钥生产管理系统

在办公自动化系统的安全设计过程中，公/私密钥对的产生与管理显得非常重要。本系统设计了一个专门生成并管理用户公/私钥对的密钥生产管理系统（the Key Build and Management，KBM）。所有办公用户的公私钥对都由 KBM 创建和管理，KBM 能将生成的公私密钥对导出到文件中，并使用相应用户的登录口令对其进行加密，以此来保证只有知道口令的用户才能解密并使用其密钥文件。由于 KBM 需要向用户提供签名真实性的验证信息，因此在 KBM 子系统中保存了用户—公钥之间的一个映射关系，当某个文档的接收者 R 需要验证签名的真实性时，他只需将文档上所声明的用户签名输入给 KBM 子系统，则 KBM 返回相应用户的公钥，由于用户的公/私密钥均是由 KBM 创建和管理的，与用户没有关系，因此 R 可以信任 KBM 返回的公钥，并使用该公钥来验证签名的真实性，当用户 S 创建了文档并需要对文档签名时，KBM 系统将提示其输入登录名以及密码，验证通过后，KBM 将该用户的私钥作为请求结果返回给他，否则返回非法访问。

图 1 显示了文档创建者与接收者和 KBM 之间进行交互的过程。

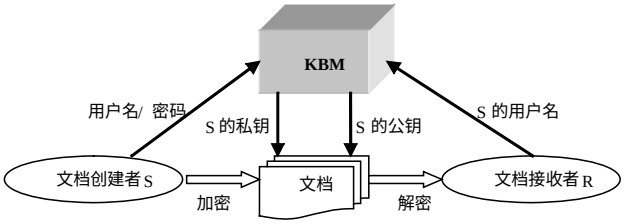

图 1 密钥生产管理系统示意图

在办公自动化系统中，很难假定其使用用户的数量，当用户数量很多时，KBM 子系统很可能成为整个系统运行效率的瓶颈。因为 KBM 集中管理所有用户的公/私密钥对，而用户请求签名以及请求签名验证又是非常频繁的操作，因此很可能导致某个用户在向 KBM 请求签名密钥时由于等待队列过长在很长的一段时间内得不到响应，但如果将密钥分散管理，又使得安全性要求得不到保证，因此需要改进 KBM 的结构，使得它可以满足大访问量、快速反应的需求。KBM 反应迟钝的根本原因在于对每一个用户都单独使用了一个密钥文件，这种方式的一个优点就是查找速度快，因为可以以用户名作为密钥文件名，当用户请求到达时，通过其用户名即可直接打开相应的密钥文件；其缺点就是对每一个用户的请求都需要打开并关闭一次文件，执行一次 I/O 操作，而这种操作是相当费时的。在这里所作的改进就是通过设置一个缓冲区来减少系统可能需要的 I/O 次数，缓冲区的大小需要根据实际硬件系统的配置来确定，因此对其的实现是动态可配置的。当某个用户的密钥是第 1 次请求时，KBM 需要执行读文件操作，并将结果缓存到缓冲区中，下一次对同一个用户的请求即可从缓冲区中获得结果。因此 KBM 查找的顺序是先从缓冲区中搜索，然后再搜索外存。在缓冲区中使用高效的 Hash 搜索方法来减少在缓冲中的搜索时间。再进一步的改进方法可以使用多线程机制，由于密钥的请求都属于读操作，使用多个线程不会造成数据间的不一致，因此这里采用线程池的技术来进一步提高 KBM 的运行效率。线程池中最大线程的数量因为关系到硬件资源所以也是以动态可配置的方式来实现，对每一个用户的请求，首先从线程池中获得一个线程，如果没有空闲线程对象且线程数量还没有达到上限则生成一个，然后使用该线程来完成用户的请求。图 2 表示了经改进后 KBM 的内部结构。

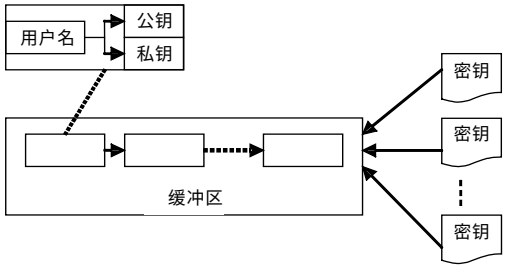

图 2 KBM 内部结构示意图

5 系统实现

我们的办公自动化系统是使用 Lotus Notes 平台来开发的，Notes 本身提供了数字签名的功能，但从实际应用的角度来讲，只设置这一层的安全保护是不够的，因此在它的基础上实现了自己的数字签名方法，通过双重加密来加强系统的安全性。

当阅读者打开签名的文档时，首先进行基于 Notes 本身的数字签名验证。Notes 签名验证完毕，将调用表单 PostOpen 事件函数，PostOpen 事件函数将检查本文档是否存在双重签名，如果存在，将调用我们专门编写的数字签名验证程序，它根据签名者标识从 KBM 子系统中取得签名者的公钥，然后对签名结果进行验证，验证结果将通过对话框显示。

图 3 表示的是双重数字签名的过程。

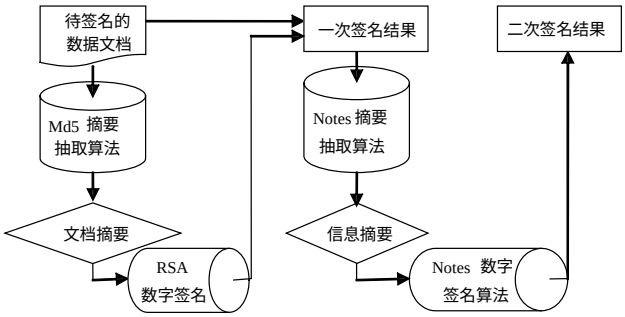

图 3 双层数字签名创建过程

图 4 表示的是如何对双重数字签名的文档进行验证。

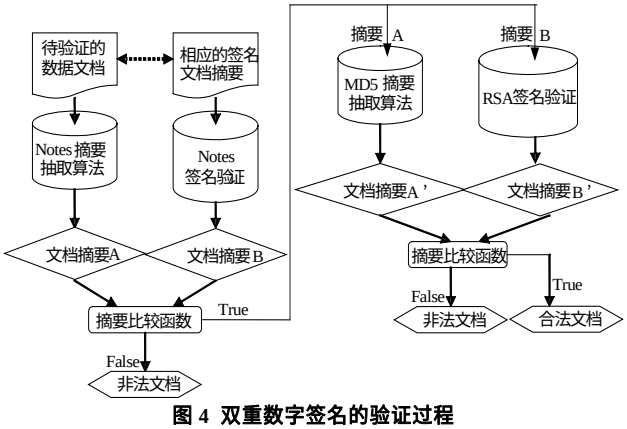

图 4 双重数字签名的验证过程

6 数字签名的效率测试

在数字签名和认证测试过程中，我们采用了 P4 1.8GHz，256MB 内存的微机作测试，结果见表 1~表 3。

（下转第 51 页）

关系进行转化。以 C_1 为例，同时有 B_1 和 B_2 向它提出服务请求。这时，把 C_1 看作两个不同的节点。经过这样的转化之后，可以把网状模型用一棵树来表示。其中，树根是用户实体，树的叶子是终端节点。

在最坏情况下，设每一级都有 x 个服务实体，共有 n 级。则形成了一棵深度为 n－1 的满 x 叉树。图 5 是 x＝3 的情况。

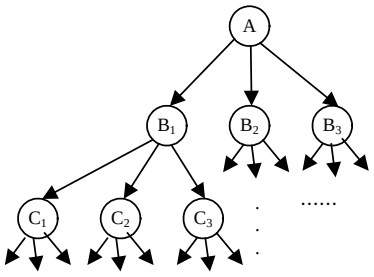

图 5 服务关系树

由于在服务供应链的确立阶段，信息交互的次数是 $O(n)$，相对于循环询问阶段来说，该复杂度是相当小的，因此这里主要讨论循环询问阶段的复杂度。

在询问阶段共需信息交互的次数为

$$x^1 + \cdots + x^n = \sum_{i=1}^{n} x^i$$

此时复杂性为 $O(x^n)$。可见，理论上在最坏情况下的复杂度是相当高的。但在实际情况下，$n \le 5$，所以这样的复杂性还是可行的。

进一步为了降低整个建立过程中信息交互的次数，特别是询问阶段的交互次数，我们在服务发现的时候可以适当地引入预测技术。即在服务发现的同时，对最终采纳该服务的可能性 p 进行一定的动态预测，在询问的时候就可以根据 p 的值筛选出一部分最可能的服务实体加以询问，其它的则直接丢弃。例如，在图 5 中，一开始就把 B_1、B_2 和 B_3 中最不可能被选中的实体（假设是）B_3 排除掉。这样，交互的复杂性就降低了 33%。

5 结语

本文首先给出了与服务供应链相关概念的形式化描述，然后重点讨论了从用户发出服务请求开始，到整个服务供应链初步建立的一个完整过程。最后就提出的建立过程中每个步骤采用的算法进行了复杂性分析。

需要指出的是，我们主要的讨论是在假设对于一个服务实体，它对下一级的每一个服务请求都是相同的这个基础上进行的。也就是说这里讨论的是简单服务供应链的建立过程。对于前面提到的树状的服务供应链的讨论将更为复杂。

在讨论的过程中，我们简要提到了服务评价和下级服务实体发现的概念，没有对其进行详细的分析。在今后的研究中，对这些概念将作进一步的研究与探讨。

随着研究的不断深入，我们还将对树状的服务供应链的的建立方法作进一步的研究。

参考文献

1 Ervin R. Chains of Commitment Software Architecture. ACM SIGecom Exchanges, 2002,3(1)

2 Verdicchio M, Colombetti M. Commitments for Agent-based Supply Chain Management. ACM SIGecom Exchanges, 2002,3(1)

3 石　丽，李　坚，李宏斌. 电子商务供应链中零库存的研究. CIMS, 2003, 9(5)

4 蒋白桦，王丹力，王宏安等. 从复杂性的角度分析供应链管理问题. 计算机工程与应用, 2002,38(15)

5 常良峰，王　静，黄小原. 供应链的成本模型及其优化. 系统工程, 2002,20(6)

（上接第 35 页）

表 1 文档摘要抽取效率表

文档大小(k)	抽取时间(ms)
10	31
100	141
1102	1 171
5736	5 469

表 2 RSA 密钥对的生成速度表

密钥位数(十六进制)	耗费时间(s)
200	5～8
300	15
400	15～25
599	300

表 3 数字签名和认证速度表

RSA 生成的质数的位数(十六进制)	耗费时间(s)
100	1
300	1~2

从上表可以看出，当密钥大于 400 位的十六进制数时，密钥产生的时间大约是 5min，速度比较慢，但就目前来讲，1 024 位二进制（即 128 位十六进制）就可满足要求，所以该签名系统完全可以应用在办公自动化中。

7 总结

本文在办公自动化系统的背景下讨论了一种基于密钥集中式管理的数字签名方法的实现，通过在安全性和效率之间的权衡选择了 MD5 算法作为抽取公文电子摘要的方法，经测试表明本文实现的算法具有良好的运行性能，抽取近 5MB 大小的文档摘要所需时间仅约为 5s，因此可以满足一般的办公自动化实用要求。在抽取的文档摘要上本文选择了 RSA 公开密钥算法作为数字签名的方法，该方法由于其密钥的不对称的性质可以很好地支持数字签名的功能，但其安全性是通过大数因子分解的计算复杂性来保证的，因此要提高安全性就需要选择更大的素数作为算法的输入。由于使用过大的素数会给运算效率带来显著的影响，因此 RSA 算法同样需要在安全性和效率之间找到一个平衡点。为了保证签名的真实性，本文提出了密钥集中管理及其实现方式，最后描述了在 Notes 实现的办公自动化系统中签名与签名验证的过程。

参考文献

1 (美)Schneier B. 吴世忠译. 应用密码学 :协议、算法与 C 源程序[M]. 北京：机械工业出版社, 2000

2 李育强. 密码学的进展. 中国科技大学研究生院信息安全国家重点实验室, 2000-06

3 Burnett S.How RSA Works. Engineering Brief RSA Data Security, Inc., 1977

4 Rivest R. The MD5 Message Digest Algorithm. MIT Laboratory for Comuter Science and RSA Data Security, Inc., 1992-04

Shuttleworth M. Digital Certificates for Extranet Authentication. President of Thawte Consulting, 1999-11